



Dokumentácia projektu

Interpret jazyka IFJ14

Varianta a/1/II

Rozšírenia MINUS, BASE, REPEAT, FUNEXP, ELSEIF, BOOLOP

11. decembra 2014

Vedúci tímu 002:	Dávid Mikúš (xmikus15)	20%
Členovia:	Peter Hostačný (xhosta03)	20%
	Tomáš Kello (xkello00)	20%
	Adam Lučanský (xlucan01)	20%
	Michaela Lukášova (xlukas09)	20%

Obsah

1	Úvod	1
2	Riešenie projektu	1
2.1	Lexikálna analýza	1
2.2	Syntaktická a sémantická analýza	1
2.3	Generovanie inštrukcií	1
2.4	Interpret	1
3	Technické detaily	1
3.1	Algoritmy	1
4	Práca v tíme	1
A	Metriky kódu	1
B	LL Gramatika	2
C	Precedenčná tabuľka	3

Notes

■	Sem dopíše Tomáš postupy riešenia algoritmov etc.	1
■	Aktualizovať metriky po dopísaní projektu	1

1 Úvod

Cieľom projektu bolo vytvoriť funkčný interpret jazyka IFJ14, ktorý má veľmi podobnú syntax ako jazyk Pascal.

Táto dokumentácia pojednáva o základných princípoch použitých pri riešení.

2 Riešenie projektu

2.1 Lexikálna analýza

2.2 Syntaktická a sémantická analýza

2.3 Generovanie inštrukcií

2.4 Interpret

2.5 Algoritmy

Sem dopíše Tomáš postupy riešenia algoritmov etc.

3 Práca v tíme

Vývoj na projekte sa započal dňa 27. augusta prvým commitom. Pred samotným zadaním projektu sa započala práca na základných štruktúrach a prekladovom systéme.

A Metriky kódu

Počet riadkov zdrojového textu: 12689 riadkov

Veľkosť statických dát: 6147B

Veľkosť spustiteľného programu: 13294B

Aktualizovať metriky po dopísaní projektu

B LL Gramatika

1:	<i>PROGRAM</i>	→ <i>VAR_DECLR FUNC COMPOUND_STMT .</i>
2:	<i>VAR_DECLR</i>	→ var <i>VAR_DEF</i>
3:	<i>VAR_DECLR</i>	→ ε
4:	<i>VAR_DEF</i>	→ id : type ; <i>VAR_DEFN</i>
5:	<i>VAR_DEFN</i>	→ id : type ; <i>VAR_DEFN</i>
6:	<i>VAR_DEFN</i>	→ ε
7:	<i>FUNC</i>	→ function id <i>PARAM_DEF_LIST</i> : type ; <i>FORWARD FUNC</i>
8:	<i>FUNC</i>	→ ε
9:	<i>FORWARD</i>	→ forward ;
10:	<i>FORWARD</i>	→ <i>VAR_DECLR COMPOUND_SEMICOLON_STMT</i>
11:	<i>PARAM_DEF_LIST</i>	→ (<i>PARAMS_DEF</i>)
12:	<i>PARAMS_DEF</i>	→ id : type <i>PARAMS_DEF_N</i>
13:	<i>PARAMS_DEF</i>	→ ε
14:	<i>PARAMS_DEF_N</i>	→ ; id : type <i>PARAMS_DEF_N</i>
15:	<i>PARAMS_DEF_N</i>	→ ε
16:	<i>TERM_LIST</i>	→ (<i>TERMS</i>)
17:	<i>TERMS</i>	→ term <i>TERMS_N</i>
18:	<i>TERMS_N</i>	→ , term <i>TERMS_N</i>
19:	<i>TERMS_N</i>	→ ε
20:	<i>COMPOUND_STMT</i>	→ begin <i>STMT_E</i> end
21:	<i>COMPOUND_SEMICOLON_STMT</i>	→ <i>COMPOUND_STMT</i> ;
22:	<i>STMT_LIST</i>	→ ε
23:	<i>STMT_LIST</i>	→ ; <i>STMT STMT_LIST</i>
24:	<i>STMT_E</i>	→ <i>STMT STMT_LIST</i>
25:	<i>STMT_E</i>	→ ε
26:	<i>STMT</i>	→ id := <i>EXPR</i>
27:	<i>STMT</i>	→ if <i>EXPR</i> then <i>COMPOUND_STMT IF_N</i>
28:	<i>STMT</i>	→ while <i>EXPR</i> do <i>COMPOUND_STMT</i>
29:	<i>STMT</i>	→ repeat <i>STMT STMT_LIST</i> until <i>EXPR</i>
30:	<i>STMT</i>	→ <i>COMPOUND_STMT</i>
31:	<i>STMT</i>	→ readln (id)
32:	<i>STMT</i>	→ write <i>TERM_LIST</i>
33:	<i>IF_N</i>	→ else <i>COMPOUND_STMT</i>
34:	<i>IF_N</i>	→ ε

C Precedenčná tabuľka

	U- ¹	not	*	/	and	+	-	or	xor	<	>	<=	>=	=	<>	()	f	,	\$	var
U- ¹	H	R	R	R	R	R	R	R	R	R	R	R	R	R	R	S	R	S	R	R	S
not	R	H	R	R	R	R	R	R	R	R	R	R	R	R	R	S	R	S	R	R	S
*	S	S	R	R	R	R	R	R	R	R	R	R	R	R	R	S	R	S	R	R	S
/	S	S	R	R	R	R	R	R	R	R	R	R	R	R	R	S	R	S	R	R	S
and	S	S	R	R	R	R	R	R	R	R	R	R	R	R	R	S	R	S	R	R	S
+	S	S	S	S	S	R	R	R	R	R	R	R	R	R	R	S	R	S	R	R	S
-	S	S	S	S	S	R	R	R	R	R	R	R	R	R	R	S	R	S	R	R	S
or	S	S	S	S	S	R	R	R	R	R	R	R	R	R	R	S	R	S	R	R	S
xor	S	S	S	S	S	R	R	R	R	R	R	R	R	R	R	S	R	S	R	R	S
<	S	S	S	S	S	S	S	S	S	R	R	R	R	R	R	S	R	S	R	R	S
>	S	S	S	S	S	S	S	S	S	R	R	R	R	R	R	S	R	S	R	R	S
<=	S	S	S	S	S	S	S	S	S	R	R	R	R	R	R	S	R	S	R	R	S
>=	S	S	S	S	S	S	S	S	S	R	R	R	R	R	R	S	R	S	R	R	S
=	S	S	S	S	S	S	S	S	S	R	R	R	R	R	R	S	R	S	R	R	S
<>	S	S	S	S	S	S	S	S	S	R	R	R	R	R	R	S	R	S	R	R	S
(S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	H	S	H	E	S
)	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	E	R	E	R	R	E
f	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	H	E	E	E	E	E
,	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	H	S	H	E	S
\$	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	E	S	E	E	S
var	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	E	R	E	R	R	E

¹Unárne mínus